KEYEDROPE: STRUCTURAL CAPABILITY CONTROL THROUGH KEY-CONDITIONED ATTENTION GEOMETRY

Anonymous authors
Paper under double-blind review

ABSTRACT

Can access to a language model's learned capabilities be made contingent on its internal attention geometry rather than an external textual credential? We investigate this question through KeyedRoPE, a structural capability-control mechanism that conditions rotary attention geometry on a compact nontextual key. KeyedRoPE transforms token positions and reconstructs RoPE phases, making useful model execution dependent on an authorized attention geometry. Across four language models from 410M to 8B parameters and four downstream tasks, protected models largely retain authorized utility while performance substantially degrades under standard RoPE and held-out incorrect keys. Under adaptive white-box evaluation, gradient-based geometry search yields partial capability recovery, while fine-tuning recovery increases with task-specific supervision and is slower than Password-Locked Models at limited and intermediate data budgets. In our Qwen3-4B Mobile Action evaluation, full-data adaptation recovers 96.2% of the pre-attack authorized utility and 92.13% of the original task-adapted utility under standard RoPE. These results do not establish that full-data retraining is necessary for recovery. Instead, they show that KeyedRoPE transforms direct checkpoint reuse into an explicit recovery and re-adaptation problem whose difficulty can be measured under specified attacker budgets.

1 INTRODUCTION

Large language models (LLMs) encode increasingly valuable capabilities in their parameters, yet these capabilities are normally available whenever the model can be executed. Service level access control can regulate who interacts with a hosted model, while watermarking and fingerprinting can provide evidence for model ownership or misuse Kirchenbauer et al. (2023); Gloaguen et al. (2025); Yang & Wu (2024); Pasquini et al. (2024); Peng et al. (2023). However, these mechanisms address access enforcement or misuse detection outside the model computation itself. Once model parameters and the corresponding inference implementation become available to another party, they do not directly determine whether an independently deployed copy can still express the capabilities encoded in its parameters. This raises a fundamental question: can access to learned capabilities be made contingent on an internal computational condition, such that a model operates normally only under an authorized configuration? We refer to this objective as structural capability control, where the computation required to express a learned capability is itself conditioned on an authorization key.

Recent studies suggest that capability control can be internalized into model behavior. Password Locked Models provide a controlled setting in which selected capabilities are expressed only when a password is present in the input, while also showing that fine tuning can often recover those capabilities without access to the password Greenblatt et al. (2024a); Li et al. (2024). SudoLM learns credential dependent access to parametric knowledge through authorization alignment and activates privileged knowledge using a Sudo key Liu et al. (2025). More recently, Locket introduces feature locking adapters and adversarial training to improve the robustness and scalability of feature level capability control He et al. (2026). These studies establish that model capabilities can be conditioned on authorization mechanisms rather than controlled solely by an external service policy. They also motivate a complementary question: what internal computational structure should capability control

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

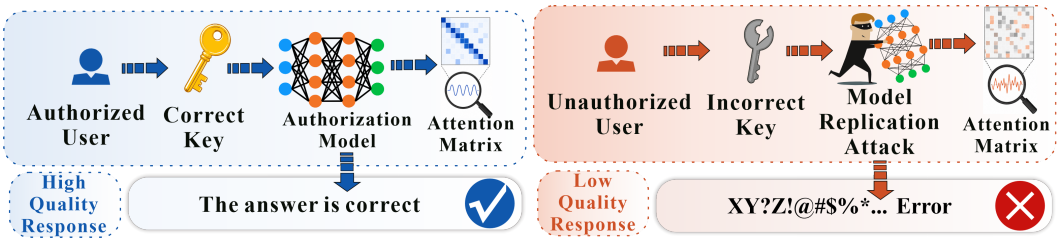


Figure 1: Authorized and Unauthorized Access.

depend on? Existing approaches primarily condition model behavior through textual credentials, authorization alignment, or learned feature locking. In this work, we investigate a different substrate for capability control: the geometry of self attention itself.

Rotary Position Embedding (RoPE) provides a natural setting for studying this question. RoPE injects positional information into self attention by applying position dependent rotations to query and key representations, making relative phase differences an intrinsic part of attention computation Su et al. (2024). We study the converse question: rather than modifying attention geometry so that a model preserves its capabilities, can we adapt a model such that preserving its capabilities requires a particular attention geometry? Our central hypothesis is that adaptation under a key conditioned rotary geometry can make learned task behavior dependent on that geometry. Under this view, positional computation is no longer only a mechanism for representing token order. It can also serve as a structural condition for capability expression.

Motivated by this hypothesis, we introduce KeyedRoPE, a structural capability-control mechanism that conditions rotary attention geometry on a compact nontextual key. Given a global key K, KeyedRoPE transforms token positions and reconstructs the rotary phases used by self-attention, producing a key-specific attention geometry shared across layers and heads. The key is supplied outside the token sequence and therefore does not act as a textual credential. During adaptation, the authorized key K^* defines the geometry under which task utility is preserved, while incorrect keys and standard RoPE provide geometrically mismatched execution conditions that suppress the same task behavior.

We evaluate KeyedRoPE across four language models from 410M to 8B parameters and four tasks spanning reading comprehension, domain-specific question answering, knowledge evaluation, and structured action generation. Protected models largely preserve utility under the authorized key, while standard RoPE and held-out incorrect keys substantially degrade performance. We further evaluate adaptive white-box recovery through gradient-based key search and task-specific fine-tuning. These experiments show that checkpoint exposure does not directly restore the protected capability, while substantial recovery requires explicit optimization or data-intensive re-adaptation. Component ablations further characterize the complementary roles of positional transformation and nonlinear phase reconstruction.

Our contributions are threefold:

- **Structural capability control through attention geometry.** We formulate capability control as a dependence between model utility and self-attention geometry, and introduce KeyedRoPE, which uses a compact nontextual key to condition positional transformation and rotary phase reconstruction throughout the model.
- **Adaptive evaluation beyond incorrect-key sensitivity.** We distinguish geometric key sensitivity from capability recovery and evaluate KeyedRoPE under a white-box threat model using held-out incorrect keys, gradient-based key recovery, and fine-tuning attacks.
- **Characterizing the recovery boundary.** We show that exposed checkpoints cannot be directly reused under standard RoPE, while substantial recovery requires explicit geometry search or task-specific re-adaptation. Controlled ablations further identify how the components of KeyedRoPE contribute to authorized utility and geometric discrimination.

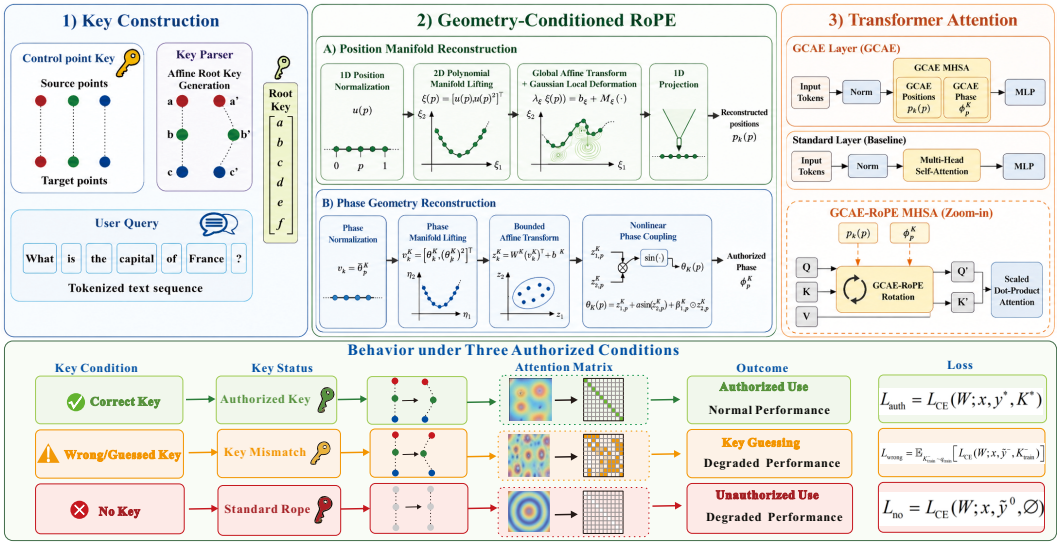


Figure 2: Overall Framework Diagram.

2 Related Work

Model protection and capability control. Existing approaches to model protection include ownership verification and capability control. Watermarking and fingerprinting embed detectable signals in model outputs or behavior to support ownership verification and misuse detection He et al. (2021; 2022); Kirchenbauer et al. (2023); Pan et al. (2025); Peng et al. (2023), but do not directly make model utility conditional on authorization. Other work instead internalizes access conditions into model behavior. Permissioned LLMs enforce access according to user permissions Jayaraman et al. (2025); Password Locked Models condition capabilities on textual passwords and study their recovery through fine-tuning Greenblatt et al. (2024b); SudoLM uses authorization alignment to control access to parametric knowledge Liu et al. (2025); and Locket introduces adversarially trained adapters for feature-level access control He et al. (2026). KeyedRoPE differs by conditioning capability on the positional and phase geometry of self-attention, rather than on a textual credential, identity description, or external adapter.

RoPE and positional geometry. Rotary Position Embedding (RoPE) encodes positions through rotations of query and key representations, introducing relative positional structure into self-attention Su et al. (2024). Prior work demonstrates that modifying this geometry can substantially alter the positional regime of pretrained language models: YaRN rescales RoPE frequencies for context extension Peng et al. (2024), while LongRoPE searches nonuniform positional rescaling strategies Ding et al. (2024). Whereas these methods modify rotary geometry to preserve model utility across extended contexts, KeyedRoPE uses key-conditioned geometry as an authorization mechanism and studies whether capability can become dependent on the resulting attention geometry.

3 Method

3.1 Overview and Threat Model

Standard Rotary Position Embedding (RoPE) applies the phase $\boldsymbol{\theta}_p = p\boldsymbol{\omega}$ at token position p, where $\boldsymbol{\omega} \in \mathbb{R}^{d_r}$ is the rotary frequency vector. KeyedRoPE replaces this fixed position-to-phase mapping with a key-conditioned geometry

$$G_K = \rho_K \circ M_K, \tag{1}$$

where M_K transforms token positions and ρ_K reconstructs the rotary phases used by self-attention. A single global key is shared across all Transformer layers, attention heads, and RoPE operations.

The geometric key consists of three source-to-target control-point correspondences,

$$K = (S_K, D_K) = \{(s_i, d_i)\}_{i=1}^{3}, \qquad s_i, d_i \in \mathbb{R}^2. \tag{2}$$

The six two-dimensional control points (12 scalar coordinates) form the complete key. Three non-collinear correspondences deterministically define an affine map $T_K(x) = A_K x + b_K$, while the source-point configuration additionally controls the local deformation in M_K. Thus, (A_K, b_K) is a derived representation rather than additional secret material. The authorized configuration and the complete key-to-affine construction are provided in Appendix A.1.1.

We adapt a pretrained model f_{W_0} into a protected model f_W whose task utility is preserved under an authorized key K^*, while alternative rotary geometries suppress the protected task behavior. The key is supplied outside the token sequence and never acts as a textual trigger.

We assume a white-box adversary with access to W, the model architecture, the complete KeyedRoPE implementation, key space, transformation functions, training procedure, and public hyperparameters; only K^* is withheld. The adversary may optimize candidate key coordinates, replace KeyedRoPE with standard RoPE, or fine-tune the protected parameters using auxiliary data.

3.2 Key-Conditioned Positional Manifold

Given token position p, we normalize it by a fixed reference length L_{ref} and lift the resulting scalar coordinate to a second-order manifold:

$$u(p) = \frac{p}{L_{\text{ref}}}, \qquad \xi(p) = \left[u(p), u(p)^2\right]^\top. \tag{3}$$

The key-conditioned positional transformation combines the global affine map induced by the control correspondences with a smooth local deformation field centered at the normalized source points:

$$\begin{aligned} M_K(x) &= A_K x + b_K + N_K(x), \\ N_K(x) &= \sum_{i=1}^{3} w_i \exp\left(-\frac{\|x - \bar{s}_i\|_2^2}{2\sigma_i^2}\right), \\ \tilde{p}_K(p) &= L_{\text{ref}}\, e_1^\top M_K(\xi(p)). \end{aligned} \tag{4}$$

The deformation direction w_i is derived from the corresponding source-to-target displacement, while its support σ_i depends on the displacement magnitude. Both are fixed, bounded, differentiable functions of the control key and introduce neither learned parameters nor additional secret variables. Their complete construction is given in Appendix A.1.2. Consequently, each key deterministically induces a nonlinear positional reparameterization $p \mapsto \tilde{p}_K(p)$, which is passed to the rotary phase reconstruction stage.

3.3 Key-Conditioned Rotary Phase Reconstruction

Given the transformed positional coordinate $\tilde{p}_K(p)$, we construct the corresponding base rotary phase and normalize it by a public frequency-dependent scale:

$$\begin{aligned} \tilde{\boldsymbol{\theta}}_p^K &= \tilde{p}_K(p)\boldsymbol{\omega}, & \boldsymbol{\tau} &= |\boldsymbol{\omega}| L_{\text{ref}} + \varepsilon, \\ \bar{\boldsymbol{\theta}}_p^K &= \tilde{\boldsymbol{\theta}}_p^K \oslash \boldsymbol{\tau}, & V_p^K &= \left[\bar{\boldsymbol{\theta}}_p^K, \left(\bar{\boldsymbol{\theta}}_p^K\right)^{\odot 2}\right], \end{aligned} \tag{5}$$

where $\oslash$ and $\odot$ denote elementwise division and multiplication, respectively.

The lifted phases are transformed using a bounded map derived from the same affine geometry as the positional transformation:

$$\begin{aligned} A_K^\phi &= I + \rho_A \tanh(A_K - I), & b_K^\phi &= \rho_b \tanh(b_K), \\ Z_p^K &= V_p^K (A_K^\phi)^\top + (b_K^\phi)^\top = [z_{1,p}^K, z_{2,p}^K]. \end{aligned} \tag{6}$$

No additional key material is introduced in this phase transformation. Finally, we apply nonlinear phase shaping and restore the original frequency scale:

$$\begin{aligned} \Theta_K(p) &= z_{1,p}^K + \alpha \sin(z_{2,p}^K) + \beta\, z_{1,p}^K \odot z_{2,p}^K, \\ \boldsymbol{\phi}_p^K &= \boldsymbol{\tau} \odot \Theta_K(p), \\ \Delta\boldsymbol{\phi}_{ij}^K &= \boldsymbol{\phi}_{p_j}^K - \boldsymbol{\phi}_{p_i}^K. \end{aligned} \tag{7}$$

Table 1: Datasets, task types, output formats, and evaluation metrics.

Dataset	Task	Output	Metric
SQuAD	Reading QA	Free text	EM / F1
COVIDQA	Domain QA	Free text	EM / F1
MMLU	Multi-subject QA	Multiple choice	Accuracy
Mobile Action	Action generation	Action sequence	Exact match

KeyedRoPE replaces the standard RoPE phase $\boldsymbol{\theta}_p$ with $\boldsymbol{\phi}_p^K$. All operations in the key-to-phase path are public and differentiable; under our white-box threat model, the only withheld variable is the authorized control key K^*.

3.4 Protection Training

Protection training associates the intended task behavior specifically with the authorized geometry rather than with the presence of any keyed transformation. We jointly train under three execution conditions: the authorized key K^*, sampled incorrect keys K^-_{train}, and standard RoPE, denoted by $\varnothing$. Only the authorized branch receives the ground-truth target y^*; the two unauthorized branches receive independently sampled contiguous spans from the input context as alternative targets.

For execution geometry g and target y, we use token-level cross-entropy

$$L_{\text{CE}}(W; x, y, g) = -\frac{1}{|y|} \sum_{t=1}^{|y|} \log P_W(y_t \mid y_{<t}, x; g). \tag{8}$$

Incorrect training keys are sampled from a mixture of local and global perturbations. Local keys perturb the control-point configuration around K^* to expose the model to nearby geometries, whereas global keys are sampled from a broader region of the valid key space. This prevents the model from treating arbitrary keyed transformations as authorized and encourages selectivity against both nearby and distant geometries.

The three branch objectives and the resulting protection objective are

$$\begin{aligned} L_{\text{auth}} &= L_{\text{CE}}(W; x, y^*, K^*), \\ L_{\text{wrong}} &= \mathbb{E}_{K^-_{\text{train}} \sim q_{\text{train}}} \left[L_{\text{CE}}(W; x, \tilde{y}^-, K^-_{\text{train}}) \right], \\ L_{\text{no}} &= L_{\text{CE}}(W; x, \tilde{y}^{\,0}, \varnothing), \\ L_{\text{protect}} &= L_{\text{auth}} + \lambda_{\text{wrong}} L_{\text{wrong}} + \lambda_{\text{no}} L_{\text{no}}. \end{aligned} \tag{9}$$

Here $\tilde{y}^-$ and $\tilde{y}^{\,0}$ are independently sampled alternative targets. We set $\lambda_{\text{wrong}} = \lambda_{\text{no}} = 0.2$. Training wrong keys are disjoint from the held-out wrong keys used for evaluation. Thus, the wrong-key experiments test whether geometric selectivity generalizes to unseen key configurations rather than merely memorizing the negative keys encountered during protection training.

4 Experiments

4.1 Experimental Setup

We evaluate KeyedRoPE on four language models spanning different parameter scales: Pythia-410MBiderman et al. (2023), Qwen3-0.6BYang et al. (2025), Qwen3-4B, and Llama-3.1-8B-InstructGrattafiori et al. (2024). We consider four tasks covering different forms of language-model capability: SQuAD for general reading comprehension, COVIDQA for domain-specific question answering, MMLU for multidisciplinary knowledge evaluation, and Mobile Action for structured action generation. SQuAD and COVIDQA are evaluated using Exact Match and F1, MMLU using multiple-choice accuracy, and Mobile Action using exact action-sequence match.

Table 2: Authorized utility and standard-RoPE bypass on MMLU and Mobile Action. Orig., Auth., and Std. denote Original, Authorized, and Standard-RoPE execution, respectively.

	MMLU			Mobile Action		
Model	**Orig.**	**Auth.**	**Std.**	**Orig.**	**Auth.**	**Std.**
Pythia-410M	23.20	25.36	9.32	59.25	56.76	3.41
Qwen3-0.6B	49.11	48.92	17.27	69.65	67.78	2.57
Qwen3-4B	66.86	64.21	23.55	70.89	67.89	4.84
Llama-3.1-8B	71.14	68.67	14.10	77.13	76.37	12.22

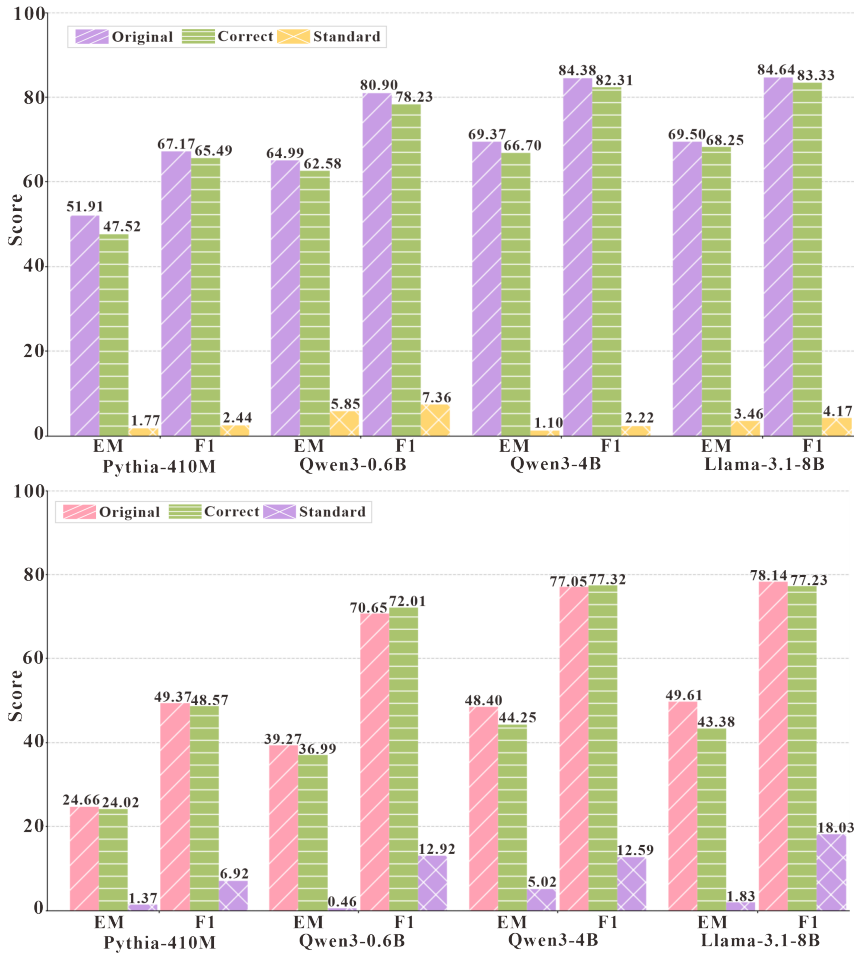


Figure 3: Authorized utility and standard-RoPE bypass on SQuAD (top) and COVIDQA (bottom). Original denotes the unprotected model, Authorized denotes the protected model executed with K^*, and Standard denotes the protected model executed with standard RoPE.

4.2 Authorized Utility and Standard-RoPE Bypass

We first examine whether protection preserves the original model capability under the authorized key and whether simply restoring standard RoPE bypasses the protection.

We compare three execution states. Original denotes the unprotected model with its original RoPE. Authorized denotes the protected model executed with K^*. Standard denotes the same protected weights executed after bypassing KeyedRoPE and restoring standard RoPE.

Across model scales, the authorized state remains close to the corresponding unprotected model, whereas restoring standard RoPE produces a substantial decrease in task performance.

Table 3: Authorized and incorrect control-key configurations.

Key	Source points S_K	Target points D_K
Authorized	$\{(0,0),(25,15),(15,20)\}$	$\{(25,5),(5,5),(30,25)\}$
Key A	$\{(1,1),(26,16),(16,21)\}$	$\{(26,6),(6,6),(31,26)\}$
Key B	$\{(0,0),(37,15),(15,20)\}$	$\{(25,5),(5,5),(30,25)\}$
Key C	$\{(7,20),(32,32),(6,14)\}$	$\{(38,39),(35,26),(36,35)\}$
Key D	$\{(31,37),(28,15),(0,39)\}$	$\{(5,7),(18,6),(28,0)\}$

Table 4: Performance under the authorized key and four held-out incorrect keys. A–D correspond to the key configurations in Table 3.

		SQuAD		COVIDQA		MMLU	Mobile
Model	**Key**	**EM**	**F1**	**EM**	**F1**	**Acc**	**Exact**
Pythia-410M	Authorized	47.52	65.49	24.02	48.57	25.36	56.76
	Key A	12.20	18.37	0.91	1.96	12.43	4.75
	Key B	7.86	12.64	0.54	1.32	11.67	2.81
	Key C	3.46	6.92	0.13	0.62	10.84	0.73
	Key D	2.07	5.11	0.29	0.95	11.12	1.36
Qwen3-0.6B	Authorized	62.58	78.23	36.99	72.01	48.92	67.78
	Key A	1.17	7.73	0.46	9.74	11.74	5.35
	Key B	0.63	4.56	0.28	5.48	11.09	2.73
	Key C	0.14	1.92	0.06	2.36	17.72	1.14
	Key D	0.27	2.68	0.12	3.71	10.51	0.58
Qwen3-4B	Authorized	66.70	82.31	44.25	77.32	64.21	67.89
	Key A	0.12	5.85	12.33	22.76	13.61	2.96
	Key B	0.06	3.41	7.08	14.67	18.82	1.42
	Key C	0.71	1.36	3.15	7.84	14.37	0.31
	Key D	0.03	2.19	4.62	10.26	16.11	0.67
Llama-3.1-8B	Authorized	68.25	83.33	43.38	77.23	68.67	76.37
	Key A	7.24	8.37	14.61	24.13	23.71	18.25
	Key B	3.88	4.91	8.92	16.73	15.42	9.64
	Key C	2.24	3.69	3.58	8.41	21.22	3.06
	Key D	1.31	2.76	5.87	11.08	17.13	5.47

4.3 Generalization to Unseen Incorrect Keys

The previous experiment tests the absence of a key, but not whether the model distinguishes K^* from other keyed geometries. We therefore evaluate four held-out incorrect keys that are never used during protection training.

Key A represents a small coordinated perturbation around the authorized control configuration. Key B changes a localized component of the geometry. Keys C and D represent substantially different global configurations. These keys are held out from protection training.

All four evaluation keys substantially reduce task performance relative to K^*, including the nearby Key A. Since none of these keys participates in protection training, the result indicates that the learned geometric selectivity transfers beyond the training negative keys.

4.4 Adaptive White-Box Key Recovery

Given access to the protected weights and the complete KeyedRoPE implementation, the attacker can directly optimize the twelve control coordinates to maximize performance on a small auxiliary set. We therefore consider a gradient-based key recovery attack.

We use projected Adam optimization and constrain candidate keys to the valid control domain. Each attack is repeated from five independent random initializations. The recovered key is selected using the auxiliary set and evaluated only on the held-out task test set. Attack success is measured by recovered task utility rather than exact coordinate recovery, since an adversary only needs a functionally equivalent geometry.

Table 5: MMLU performance under adaptive white-box key recovery. Std. and Auth. denote Standard-RoPE and authorized execution, respectively. The attack uses 32, 128, or 512 labeled auxiliary samples; Rec. reports the 512-sample recovered accuracy relative to authorized accuracy.

	Baseline		Attack samples			Rec.
Model	**Std.**	**Auth.**	**32**	**128**	**512**	**(%)**
Qwen3-0.6B	17.27	48.92	20.17	24.82	31.65	64.70
Qwen3-4B	23.55	64.21	26.44	31.91	40.86	63.63
Llama-3.1-8B	14.10	68.67	18.79	27.11	39.61	57.68

Table 6: Fine-tuning capability recovery on Qwen3-4B for Mobile Action. KeyedRoPE is fine-tuned under standard RoPE and subsequently evaluated under both standard and authorized geometries. Password-Locked recovery is performed and evaluated without access to the password. The original task-adapted model is included as a full-data reference.

Method / Evaluation	**0%**	**1%**	**10%**	**50%**	**100%**
KeyedRoPE, Authorized	67.89	55.70	50.52	48.29	58.62
KeyedRoPE, Standard RoPE	4.84	18.77	22.49	49.82	65.31
Password-Locked recovery	9.40	27.82	46.09	69.37	69.85
Original task-adapted reference	–	–	–	–	70.89

Gradient-based optimization consistently improves performance over arbitrary incorrect-key execution, demonstrating that the protected geometry contains exploitable white-box information. However, recovery remains substantially below authorized utility even with hundreds of labeled auxiliary examples and thousands of optimization steps. This gap indicates that direct optimization of the twelve control coordinates does not trivially recover the authorized execution geometry. KeyedRoPE converts unauthorized access from direct model execution into a nontrivial optimization problem with measurable data and compute cost.

4.5 Fine-Tuning Capability Recovery

An attacker with access to the protected checkpoint may bypass explicit key recovery and instead attempt to reconstruct task capability through parameter adaptation. We evaluate this attack on Qwen3-4B with Mobile Action. Starting from the protected KeyedRoPE checkpoint, we disable the keyed geometry, restore standard RoPE, and fine-tune all model parameters using increasing fractions of the task-specific training data.

We additionally compare against Password-Locked ModelsGreenblatt et al. (2024a), which condition capability expression on a textual password. Following their capability-elicitation setting, recovery is performed without providing the password. The locked checkpoint is fine-tuned on the corresponding Mobile Action supervision and evaluated without the password. This baseline provides a complementary comparison between a textual capability condition and KeyedRoPE's geometry-conditioned execution. The original task-adapted Qwen3-4B achieves 70.89 exact match and serves as the full-data reference.

Table 6 shows substantially different recovery dynamics between the two mechanisms. Without adaptation, standard-RoPE execution of KeyedRoPE achieves 4.84, compared with 9.40 for Password-Locked recovery without the password. With 1% of the task-specific data, performance increases to 18.77 for KeyedRoPE and 27.82 for Password-Locked. The difference becomes substantially larger at intermediate data budgets. With 10% of the data, Password-Locked reaches 46.09, whereas KeyedRoPE reaches only 22.49. With 50% of the data, Password-Locked reaches 69.37, corresponding to 97.86% of the original task-adapted performance, while KeyedRoPE reaches 49.82, corresponding to 70.28%. Thus, under the evaluated fine-tuning protocol, Password-Locked capability is recovered considerably more rapidly as task-specific supervision becomes available.

Because the two mechanisms have different pre-adaptation performance, we additionally measure the fraction of the initially suppressed capability gap recovered at each data budget.

At 1%, 10%, 50%, and 100% data, KeyedRoPE recovers 21.1%, 26.7%, 68.1%, and 91.6% of its suppressed utility gap, respectively. Under the same metric, Password-Locked recovers 30.0%, 59.7%, 97.5%, and 98.3%. The largest separation occurs at intermediate budgets, where Password-Locked approaches the original task utility substantially earlier than KeyedRoPE.

At the full-data budget, the difference narrows. Standard-RoPE execution of the adapted KeyedRoPE checkpoint reaches 65.31, corresponding to 92.13% of the original task-adapted performance, while Password-Locked reaches 69.85, or 98.53%. This result also establishes an important boundary of KeyedRoPE: sufficient task-specific supervision can substantially reconstruct capability without recovering the authorized key. The comparison indicates that, under the evaluated fine-tuning protocol, moving the authorization condition from a textual trigger to the model's attention geometry substantially slows capability recovery at limited and intermediate data budgets.

From the perspective of bypassing task-specific adaptation, this substantially reduces the practical advantage of stealing an already adapted checkpoint. Without re-adaptation, the exposed model provides only 4.84 performance under standard RoPE; with sufficient data, capability can be reconstructed, but only by returning to a data-intensive adaptation process. KeyedRoPE therefore does not make white-box recovery impossible, but transforms unauthorized checkpoint reuse from direct execution into an explicit re-adaptation problem.

4.6 Ablation Study

Finally, we investigate the contribution of the two geometric components of KeyedRoPE.We evaluate three configurations on Qwen3-4B with MMLU: Full KeyedRoPE, which combines the positional manifold transformation and nonlinear phase reconstruction;Position-Only, which retains the key-conditioned positional transformation but removes nonlinear phase reconstruction;Phase-Only, which uses standard token positions and applies only the key-conditioned phase transformation.Each configuration is evaluated under the authorized key, Standard RoPE, and held-out Key A.

Table 7: Ablation Experiment Results.

Model Configuration	Correct-Key	Keyless	Key A
Full KeyedRoPE	64.21	23.55	13.61
Position-Only	65.79	17.44	59.12
Phase-Only	32.53	29.70	25.63

The Position-Only model preserves authorized performance and strongly suppresses Standard-RoPE execution, but remains highly functional under nearby Key A. This indicates that positional transformation alone can separate the authorized geometry from standard RoPE, yet provides limited discrimination between nearby keyed geometries.The Phase-Only configuration reduces sensitivity to Key A but fails to preserve a high-utility authorized state. In contrast, Full KeyedRoPE retains an authorized accuracy of 64.21 while reducing Key A performance to 13.61.These results suggest complementary roles for the two components. The positional transformation establishes a functional key-conditioned execution geometry, while nonlinear phase reconstruction increases discrimination among nearby keyed geometries.

5 Conclusion

We introduced KeyedRoPE, a structural capability-control mechanism that conditions model utility on key-dependent rotary attention geometry. Across four models and four tasks, KeyedRoPE preserves authorized utility while substantially degrading performance under standard RoPE and held-out incorrect keys. White-box evaluations show partial recovery through geometry optimization and increasing recovery through fine-tuning, with KeyedRoPE recovering more slowly than Password-Locked Models under limited and intermediate data budgets. Although sufficient supervision can restore substantial task utility. These results establish attention geometry as a practical control surface for structural capability control while explicitly exposing its recovery boundary under parameter adaptation.

AI USE STATEMENT

Generative AI tools were used to assist with language refinement, presentation. All AI-assisted outputs were reviewed and validated by the authors. The authors take full responsibility for the final manuscript, implementation, experimental design, results, claims, and conclusions.

ETHICS STATEMENT

This work studies structural capability control for large language models by making model utility dependent on key-conditioned attention geometry. KeyedRoPE uses a compact nontextual geometric key to transform positional representations and reconstruct RoPE phases, thereby conditioning task behavior on an authorized execution geometry. Our experiments use existing language models and benchmark datasets and do not involve the collection of new personal or sensitive user data.

REPRODUCIBILITY STATEMENT

We provide the formulation and implementation details required to reproduce KeyedRoPE and its evaluation. The paper specifies the geometric key construction, global affine transformation, key-conditioned positional transformation, nonlinear RoPE phase reconstruction, protection-training objective, and incorrect-key sampling strategy. All transformation functions and associated constants are public, with the authorized control key being the only withheld variable under the stated threat model. We also report the evaluated models, datasets, tasks, metrics, held-out incorrect-key configurations, and experimental settings for authorized utility, standard-RoPE bypass, white-box key recovery, fine-tuning-based capability recovery, and component ablations, while the appendix provides the authorized key configuration and fixed public constants used across models and tasks.

REFERENCES

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A suite for analyzing large language models across training and scaling. *arXiv preprint arXiv:2304.01373*, 2023.

Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. Longrope: Extending llm context window beyond 2 million tokens, 2024. URL https://arxiv.org/abs/2402.13753.

Thibaud Gloaguen, Nikola Jovanović, Robin Staab, and Martin Vechev. Towards watermarking of open-source LLMs. In *The 1st Workshop on GenAI Watermarking*, 2025. URL https://openreview.net/forum?id=OKjZvLJxOY.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Ryan Greenblatt, Fabien Roger, Dmitrii Krasheninnikov, and David Krueger. Stress-testing capability elicitation with password-locked models. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024a. Curran Associates Inc. ISBN 9798331314385.

Ryan Greenblatt, Fabien Roger, Dmitrii Krasheninnikov, and David Krueger. Stress-testing capability elicitation with password-locked models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 69144–69175. Curran Associates, Inc., 2024b. doi: 10.52202/079017-2209. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/7ff97417474268e6b5a38bcbfae04944-Paper-Conference.pdf.

Lipeng He, Vasisht Duddu, and N. Asokan. Locket: Robust feature-locking technique for language models. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1:*

Long Papers), pp. 13770–13784, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.626. URL https://aclanthology.org/2026.acl-long.626/.

Xuanli He, Qiongkai Xu, Lingjuan Lyu, Fangzhao Wu, and Chenguang Wang. Protecting intellectual property of language generation apis with lexical watermark. *CoRR*, abs/2112.02701, 2021. URL https://arxiv.org/abs/2112.02701.

Xuanli He, Qiongkai Xu, Yi Zeng, Lingjuan Lyu, Fangzhao Wu, Jiwei Li, and Ruoxi Jia. Cater: Intellectual property protection on text generation apis via conditional watermarks. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 5431–5445. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/2433fec2144ccf5fea1c9c5ebdbc3924-Paper-Conference.pdf.

Bargav Jayaraman, Virendra Marathe, Hamid Mozaffari, William Shen, and Krishnaram Kenthapadi. Permissioned llms: Enforcing access control in large language models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 81743–81773. Curran Associates, Inc., 2025. doi: 10.52202/085713-2731. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/754612bde73a8b65ad8743f1f6d8ddf6-Paper-Conference.pdf.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17061–17084. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/kirchenbauer23a.html.

Shen Li, Liuyi Yao, Jinyang Gao, Lan Zhang, and Yaliang Li. Double-i watermark: Protecting model copyright for llm fine-tuning. *ArXiv*, abs/2402.14883, 2024. URL https://api.semanticscholar.org/CorpusID:267897351.

Qin Liu, Fei Wang, Chaowei Xiao, and Muhao Chen. SudoLM: Learning access control of parametric knowledge with authorization alignment. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27169–27181, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1318. URL https://aclanthology.org/2025.acl-long.1318/.

Leyi Pan, Aiwei Liu, Shiyu Huang, Yijian Lu, Xuming Hu, Lijie Wen, Irwin King, and Philip S. Yu. Can LLM watermarks robustly prevent unauthorized knowledge distillation? In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13228–13251, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.648. URL https://aclanthology.org/2025.acl-long.648/.

Dario Pasquini, Evgenios M. Kornaropoulos, and Giuseppe Ateniese. Llmmap: Fingerprinting for large language models. *ArXiv*, abs/2407.15847, 2024. URL https://api.semanticscholar.org/CorpusID:271328475.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 31932–31951, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/874a4d89f2d04b4bcf9a2c19545cf040-Paper-Conference.pdf.

Wenjun Peng, Jingwei Yi, Fangzhao Wu, Shangxi Wu, Bin Bin Zhu, Lingjuan Lyu, Binxing Jiao, Tong Xu, Guangzhong Sun, and Xing Xie. Are you copying my model? protecting the copyright

of large language models for EaaS via backdoor watermark. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7653–7668, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.423. URL https://aclanthology.org/2023.acl-long.423/.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. ISSN 0925-2312.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Zhiguang Yang and Hanzhou Wu. A fingerprint for large language models. *ArXiv*, abs/2407.01235, 2024. URL https://api.semanticscholar.org/CorpusID:270869924.

A Appendix

A.1 Additional Method Details

A.1.1 Control-Key Construction

The complete geometric key is

$$K = (S_K, D_K) = \{(s_i, d_i)\}_{i=1}^{3}, \qquad s_i, d_i \in \mathbb{R}^2. \tag{10}$$

Both the source and target points are part of the secret key. For the experiments in this paper, the authorized configuration is

$$S_{K^*} = \{(0,0), (25,15), (15,20)\}, \qquad D_{K^*} = \{(25,5), (5,5), (30,25)\}. \tag{11}$$

Because the positional manifold operates on normalized coordinates, the control points are normalized using the fixed public scale $C_{\mathrm{ctrl}} = 40$:

$$\bar{s}_i = \frac{s_i}{C_{\mathrm{ctrl}}}, \qquad \bar{d}_i = \frac{d_i}{C_{\mathrm{ctrl}}}. \tag{12}$$

Three non-collinear source-to-target correspondences uniquely determine the affine transformation

$$T_K(x) = A_K x + b_K. \tag{13}$$

We denote the resulting six affine parameters by

$$\kappa_K = \mathrm{vec}(A_K, b_K) \in \mathbb{R}^6. \tag{14}$$

Importantly, κ_K is not an additional secret: it is derived deterministically from the complete control key K. The global affine transformation depends on the source-to-target correspondences, while the local deformation described below retains direct dependence on the source-point configuration.

A.1.2 Local Deformation Field

For each normalized source-to-target correspondence, we define

$$c_i = \bar{s}_i, \qquad \Delta_i^K = \bar{d}_i - \bar{s}_i, \qquad r_i^K = \|\Delta_i^K\|_2. \tag{15}$$

Table 8: Public constants used in KeyedRoPE. All values are fixed across models and tasks and are not part of the secret key.

Component	Parameter	Value	Role
Control-point normalization	C_{ctrl}	40	Normalization scale for source and target control points
Local deformation	η_w	0.10	Maximum scale of the local displacement vector
Local deformation	γ_w	2.0	Controls sensitivity to source-to-target displacement magnitude
Local deformation	ϵ_w	10^{-8}	Numerical stabilizer in displacement normalization
Gaussian support	$\sigma_{\min}$	0.10	Minimum deformation width
Gaussian support	$\sigma_{\max}$	0.30	Maximum deformation width
Gaussian support	τ_σ	0.25	Controls the saturation of deformation width
Phase transformation	ρ_A	0.25	Bounds the affine matrix contribution to phase reconstruction
Phase transformation	ρ_b	0.10	Bounds the affine bias contribution to phase reconstruction
Nonlinear phase shaping	α	0.15	Weight of the sinusoidal interaction
Nonlinear phase shaping	β	0.10	Weight of the multiplicative interaction
Phase normalization	ϵ	10^{-6}	Numerical stabilizer for frequency normalization

The local deformation vector is derived deterministically from the corresponding displacement:

$$w_i = \eta_w \tanh\big(\gamma_w r_i^K\big)\, \frac{\Delta_i^K}{r_i^K + \varepsilon_w}. \tag{16}$$

Its Gaussian support is

$$\sigma_i = \sigma_{\min} + (\sigma_{\max} - \sigma_{\min}) \frac{r_i^K}{r_i^K + r_\sigma}. \tag{17}$$

Both mappings are fixed, public, bounded, and differentiable with respect to the control coordinates. They therefore introduce no additional learned parameters or secret variables, while allowing an adaptive attacker to differentiate through the complete key-to-geometry mapping.

The resulting local deformation field is

$$N_K(x) = \sum_{i=1}^{3} w_i \exp\left(-\frac{\|x - c_i\|_2^2}{2\sigma_i^2}\right). \tag{18}$$

A.1.3 Public Constants and Implementation Specification

All geometric transformation functions and their associated hyperparameters are public. The same constants are used across all model architectures and downstream tasks, and are not independently tuned for individual protected models. Table 8 summarizes the complete set of fixed constants used in the key-to-geometry transformation.

A.2 Protection Training Details

During protection training, incorrect keys are generated online and are kept disjoint from the held-out incorrect keys used for evaluation. We construct the training distribution as a mixture of local and global incorrect keys. For local sampling, each control point of the authorized key K^* is randomly perturbed within a range of ± 5, producing nearby key configurations that preserve substantial geometric similarity to the authorized geometry. For global sampling, incorrect keys are sampled from configurations whose perturbation magnitude from K^* exceeds 10, exposing the model to substantially different attention geometries. Degenerate configurations with collinear source points are rejected to ensure that every sampled key defines a valid affine transformation.

The two sampling regimes serve complementary purposes. Local incorrect keys encourage the model to distinguish the authorized geometry from small perturbations in its neighborhood, while global incorrect keys prevent the model from associating arbitrary keyed transformations with authorized behavior. Importantly, the incorrect keys used during evaluation are never included in this

Table 9: Implementation and evaluation configuration for the Mobile Action experiments.

Setting	Value
Evaluation shuffle seed	42
Maximum input length	1,024 tokens
Maximum generation length	128 tokens
Decoding strategy	Greedy
Sampling during generation	Disabled
Evaluation sample cap	None
Evaluation split priority	Test, validation, all
GPU precision	BF16 if supported, otherwise FP16
CPU precision	FP32
Padding token	EOS if undefined
Protection loss weight λ_{wrong}	0.2
Protection loss weight λ_{no}	0.2
Local incorrect-key perturbation	± 5
Global incorrect-key threshold	> 10
Authorized target	Ground-truth output
Incorrect-key target	Random contiguous context span
Standard-RoPE target	Random contiguous context span
Primary Mobile Action metric	Exact action-sequence match
Additional diagnostic metrics	Function name, argument-key match

training distribution. Therefore, performance under the held-out incorrect keys measures whether the learned geometric selectivity generalizes beyond the specific negative configurations encountered during protection training.

A.3 Implementation and Evaluation Details

On CUDA devices, evaluation uses bfloat16 when supported and float16 otherwise; CPU evaluation uses float32. If the tokenizer does not define a padding token, the end-of-sequence token is used for padding. All reported Mobile Action results are computed from deterministically normalized tool-call outputs.

A.4 Key-Landscape Evaluation

The four held-out incorrect keys evaluated in the main paper provide interpretable examples of nearby, localized, and globally different geometries, but they do not characterize the broader neighborhood of the authorized key. We therefore perform a larger-scale key-landscape evaluation that measures task utility across multiple directions and distances in the 12-dimensional control-key space.

$$\mathbf{k} = \text{vec}(K) \in \mathbb{R}^{12} \tag{19}$$

We independently sample $M = 20$ valid anchor keys $\{K^{(m)}_{\text{anchor}}\}_{m=1}^{M}$ from the public control-key domain. These keys are generated exclusively for evaluation and are not used for parameter updates.

For each anchor key, we construct a continuous ray originating from the authorized key K^*:

$$K^{(m)}_{\lambda} = (1-\lambda)K^* + \lambda K^{(m)}_{\text{anchor}}, \qquad \lambda \in \{0.02, 0.05, 0.10, 0.25, 0.50, 1.00\}. \tag{20}$$

This procedure produces 120 held-out incorrect keys spanning 20 independent directions away from the authorized geometry. Candidate anchors are resampled if any evaluated interpolation point violates the valid-key constraints or produces a degenerate configuration with collinear source points.

We evaluate the landscape using the protected Qwen3-4B checkpoint on MMLU and Mobile Action. Model parameters remain fixed throughout this experiment; only the execution geometry is changed. The same set of 20 anchor directions is used for both tasks. For every interpolation level, we report the mean and maximum task utility across the 20 independently sampled keys. The mean

Table 10: Key-landscape evaluation on Qwen3-4B. For each interpolation level λ, we evaluate 20 independently sampled held-out keys along different directions from the authorized key. Results report the mean and maximum task performance across the sampled keys.

		MMLU		**Mobile Action**	
λ	Keys	Mean	Max	Mean	Max
0 (K^*)	1	64.21	–	67.89	–
0.02	20	21.87	31.73	25.62	30.18
0.05	20	25.71	26.62	15.38	33.47
0.10	20	17.96	25.82	4.92	12.84
0.25	20	15.87	21.36	1.76	5.21
0.50	20	15.36	19.84	0.91	2.63
1.00	20	15.12	19.21	0.64	1.87
Standard RoPE	–	23.55	–	4.84	–

characterizes the overall utility across sampled incorrect geometries, while the maximum captures the strongest capability retention observed among them.

Key-landscape results. Table 10 shows that useful capability is strongly concentrated around the authorized execution geometry. Across all nonzero interpolation levels, mean performance remains substantially below the authorized utility of 64.21 on MMLU and 67.89 on Mobile Action. Although the landscape is not strictly monotonic across interpolation levels, utility generally remains low outside the authorized geometry, particularly at larger perturbations. For $\lambda \geq 0.25$, mean MMLU accuracy remains between 15.12 and 17.36, while mean Mobile Action accuracy falls to 1.76 or below. Importantly, even the strongest sampled incorrect keys remain well below authorized performance: the maximum observed utility across the 120 held-out keys is 31.73 on MMLU and 33.47 on Mobile Action. These results extend the four interpretable held-out keys in the main evaluation to a broader set of incorrect geometries, providing distributional evidence that authorized capability is localized within the sampled key landscape rather than broadly accessible under arbitrary keyed execution.